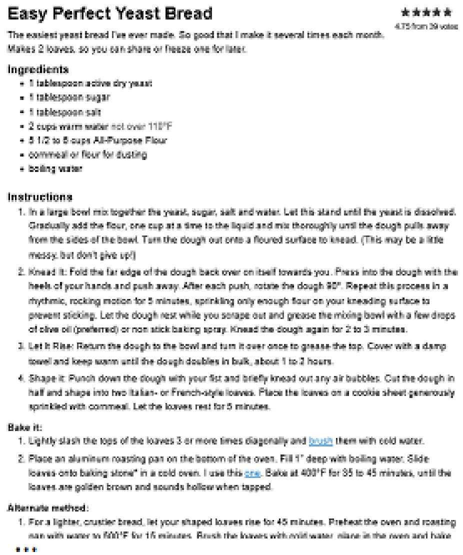
GET INTO PROGRAMMING WITH JavaScript

Axle Barr

Selection and Algorithms

Algorithm, Sequence, Selection

Its real simple, just add flour, salt, yeast and water, mix and bake
Wow, that is simple



Easy Perfect Yeast Bread

★★★★★
4.75 from 39 votes

The easiest yeast bread I've ever made. So good that I make it several times each month. Makes 2 loaves, so you can share or freeze one for later.

Ingredients

- 1 tablespoon active dry yeast
- 1 tablespoon sugar
- 1 tablespoon salt
- 2 cups warm water not over 110°F
- 5 1/2 to 6 cups All-Purpose Flour
- cornmeal or flour for dusting
- boiling water

Instructions

1. In a large bowl mix together the yeast, sugar, salt and water. Let this stand until the yeast is dissolved. Gradually add the flour, one cup at a time to the liquid and mix thoroughly until the dough pulls away from the sides of the bowl. Turn the dough out onto a floured surface to knead. (This may be a little messy, but don't give up!)
2. Knead it: Fold the far edge of the dough back over on itself towards you. Press into the dough with the heels of your hands and push away. After each push, rotate the dough 90°. Repeat this process in a rhythmic, rocking motion for 5 minutes, sprinkling only enough flour on your kneading surface to prevent sticking. Let the dough rest while you scrape out and grease the mixing bowl with a few drops of olive oil (preferred) or non stick baking spray. Knead the dough again for 2 to 3 minutes.
3. Let It Rise: Return the dough to the bowl and turn it over once to grease the top. Cover with a damp towel and keep warm until the dough doubles in bulk, about 1 to 2 hours.
4. Shape it: Punch down the dough with your fist and briefly knead out any air bubbles. Cut the dough in half and shape into two Italian- or French-style loaves. Place the loaves on a cookie sheet generously sprinkled with cornmeal. Let the loaves rest for 5 minutes.

Bake it:

1. Lightly slash the tops of the loaves 3 or more times diagonally and brush them with cold water.
2. Place an aluminum roasting pan on the bottom of the oven. Fill 1" deep with boiling water. Slide loaves onto baking stone* in a cold oven. I use this one. Bake at 400°F for 35 to 45 minutes, until the loaves are golden brown and sounds hollow when tapped.

Alternate method:

1. For a lighter, crustier bread, let your shaped loaves rise for 45 minutes. Preheat the oven and roasting pan with water to 500°F for 15 minutes. Brush the loaves with cold water, place in the oven and bake

• • •

Algorithm, Sequence, Selection

Easy Perfect Yeast Bread
The easiest yeast bread and so good

Total Time: 50 mins
Course: Bread
Cuisine: Norwegian
Servings: 2 loaves

Ingredients

- 1 tablespoon active dry yeast
- 1 tablespoon salt
- 1 tablespoon sugar
- 2 cups warm water about 60°C
- 5 1/2 to 6 cups All-Purpose Flour
- boiling water
- cornmeal or flour for dusting

Instructions

- In a large bowl, mix together the yeast, sugar, salt, and water. Leave alone until the yeast is dissolved.

Gradually add the flour, one cup at a time to the liquid and mix thoroughly until the dough pulls away from the sides of the bowl. Turn
…

Directions

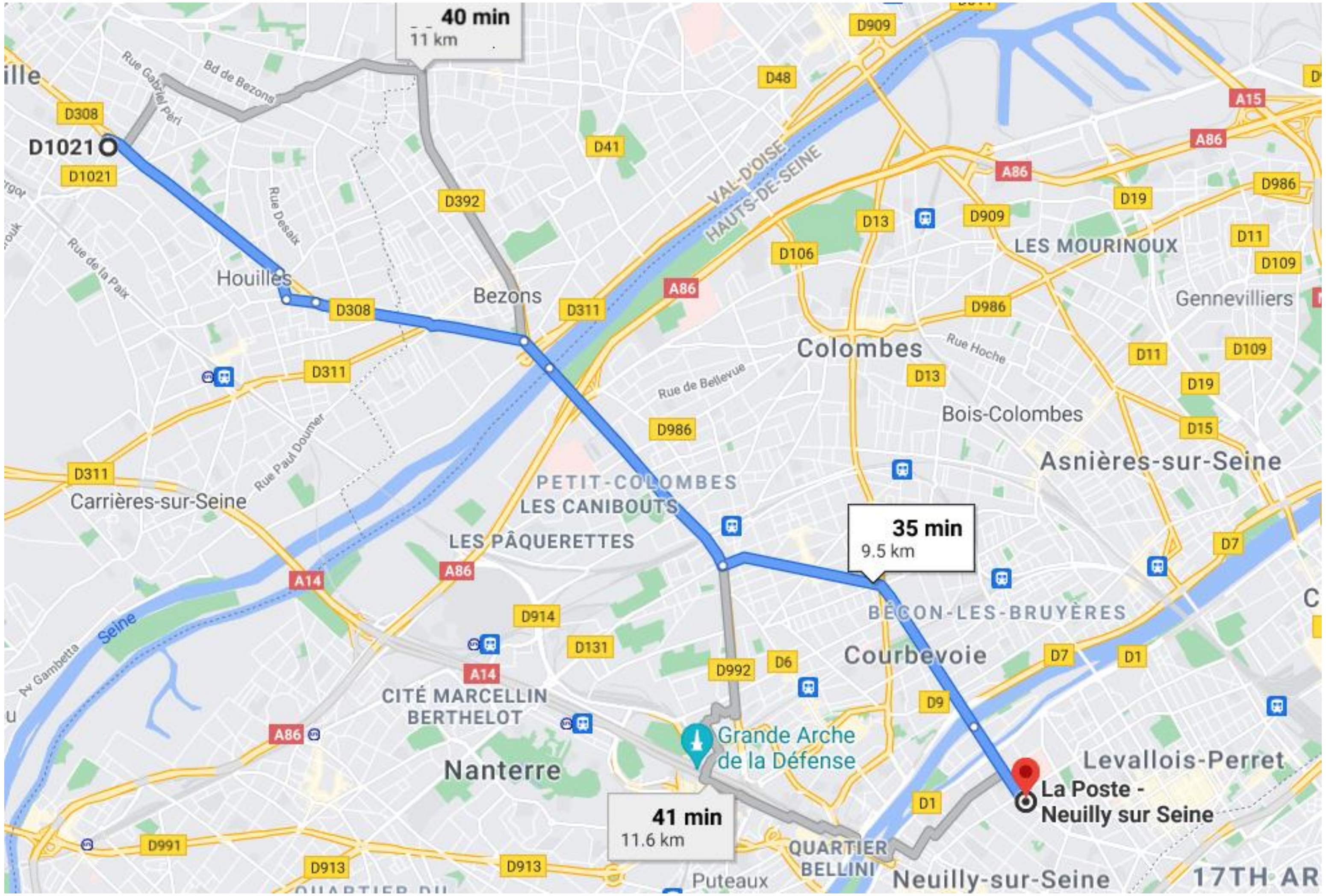
Head south for 84 m
Turn left onto Rue de la République 450 m
Turn left onto Rue Alsace Lorraine 270 m
Turn right onto Place Saint-Marc 42 m
Turn left to stay on Place Saint-Marc 35 m
Turn right onto Rue Armand Carrel 190 m
Turn left onto Quai de Paris/D6015
Continue to follow D6015 for 650 m
Continue straight onto Route de Bonsecours/D6015
Continue to follow Route de Bonsecours 650 m
Continue onto Route de Paris/D6014
Continue to follow Route de Paris 550 m
Turn left onto Sentier du Raidillon 190 m
Turn left onto Route de Paris/D914 5.0 km
At the roundabout, take the 2nd exit onto Route de Paris/D6014
…

21 Point Pre-Delivery Inspection

1. Fit all of the loose equipment supplied with the bike and tighten pedals, toe clips and straps
2. Adjust wheel quick releases and position levers correctly. Tighten wheel nuts, where fitted, and explain the importance of correct chain tension where necessary
3. Spin wheels to check trueness, then test tyre pressures.
4. Check that the saddle height and fore and aft adjustment are correctly matched to the customer.
5. Check handlebar height and handlebar angle are correctly matched to the customer.
6. Make sure that the seat post and handlebar stem are not extended further than the safety limit line.
7. Tighten saddle clip, saddle adjustment bolt, handlebar stem fixing, handlebar clamp bolt and bar ends.

…

Algorithm, Sequence, Selection



Instructions to the Machine

The statements seen so far:

- Are instructions to the computer
- They are executed one by one
- Each line is executed in a sequential order
- If a statement contains parenthesis, A_BODMAS will take precedent

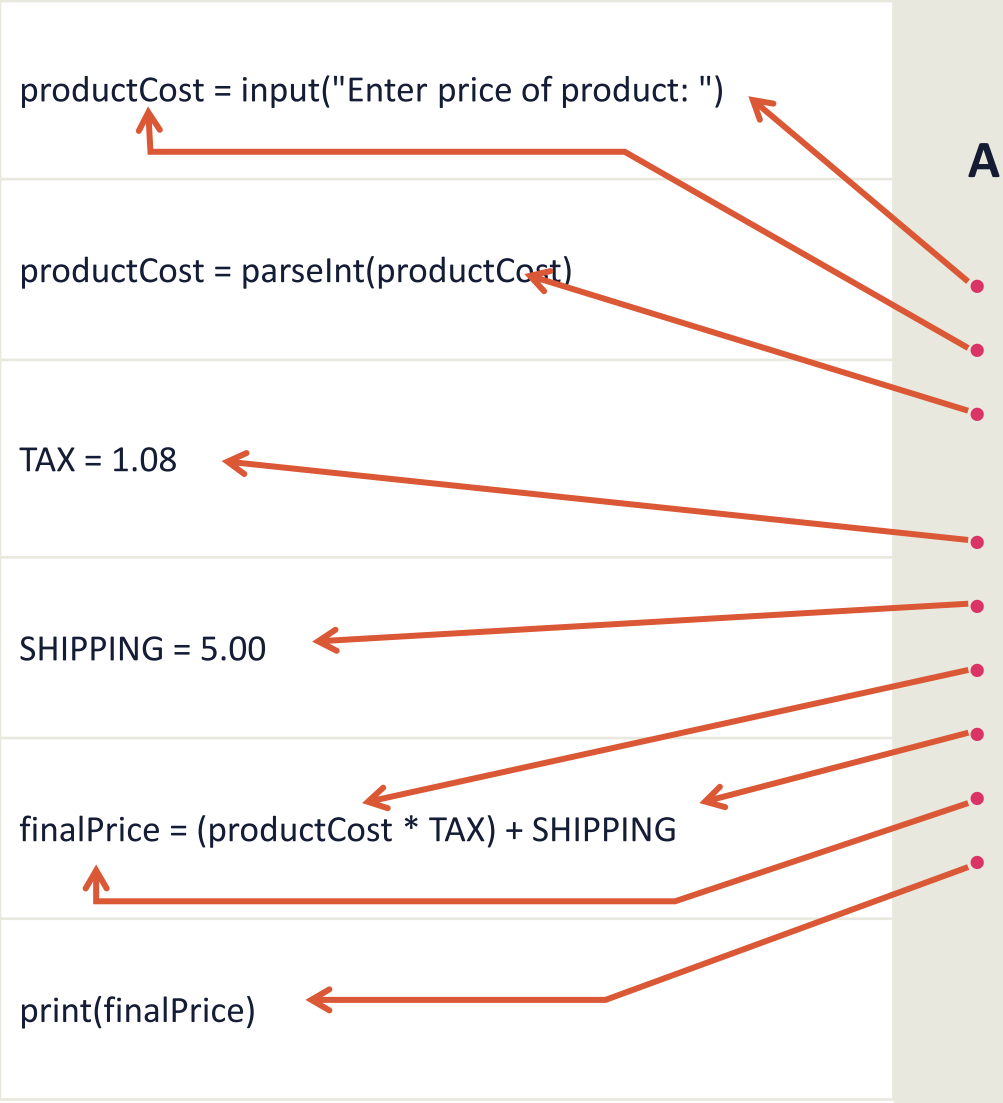
Algorithm

Steps performed by machine:

- Prompt user for product cost
- Store the user’s value in *productCost*
- Convert the value in *productCost* to a number and store it back in *productCost*
- Put tax rate into a constant variable
- Put shipping rate into a constant variable
- Multiply *productCost* by the TAX
- Add shipping cost to the product from above
- Store the new value amount in a variable called *finalPrice*
- Show *finalPrice* on the screen

Algorithm to Program (Sequencing)

Shopping Cart Program



```
productCost = input("Enter price of product: ")

productCost = parseInt(productCost)

TAX = 1.08

SHIPPING = 5.00

finalPrice = (productCost * TAX) + SHIPPING

print(finalPrice)
```

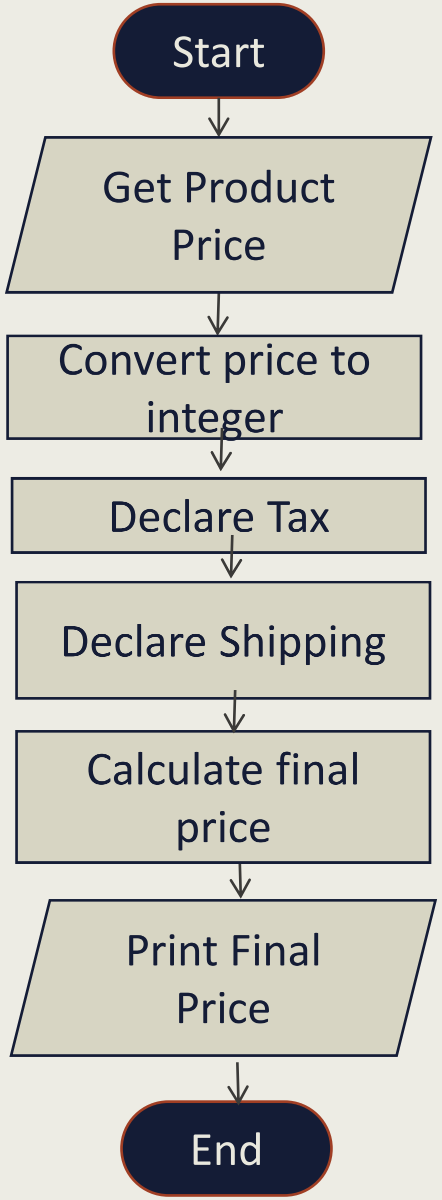
Analyzing the shopping cart program:

- Prompt user for the cost of the product they are purchasing
- Store the user’s value in **productCost**
- Convert the value in **productCost** to a number and store it back in **productCost**
- Put the tax rate into a constant variable
- Put the shipping rate into a constant variable
- Multiply **productCost** by the tax rate
- Add shipping cost to the product calculated above
- Store the calculate amount in a variable called **finalPrice**
- Show **finalPrice** on the screen

Flowchart to Program

Shopping Cart Program

```
productCost = input("Enter price of product: ")

productCost = int(productCost)

TAX = 1.08

SHIPPING = 5.00

finalPrice = (productCost * TAX) + SHIPPING

print(finalPrice)
```



Start

Get Product Price

Convert price to integer

Declare Tax

Declare Shipping

Calculate final price

Print Final Price

End

Expanding the Shopping Cart

Selection (Decisions):

- Typical decisions:
    - if it is raining, I will take an umbrella,
    - if the road is closed, I will take an alternate route,
    - if the movie is not playing, I will watch a different movie

We want to offer free shipping to certain customers
Create a separate program or start using decisions

```
productCost = prompt("Enter price of product: ");
productCost = parseFloat(productCost);
TAX = 1.08;
SHIPPING = 5.00;
finalPrice = (productCost * TAX) + SHIPPING;
printOut  = finalPrice;
```

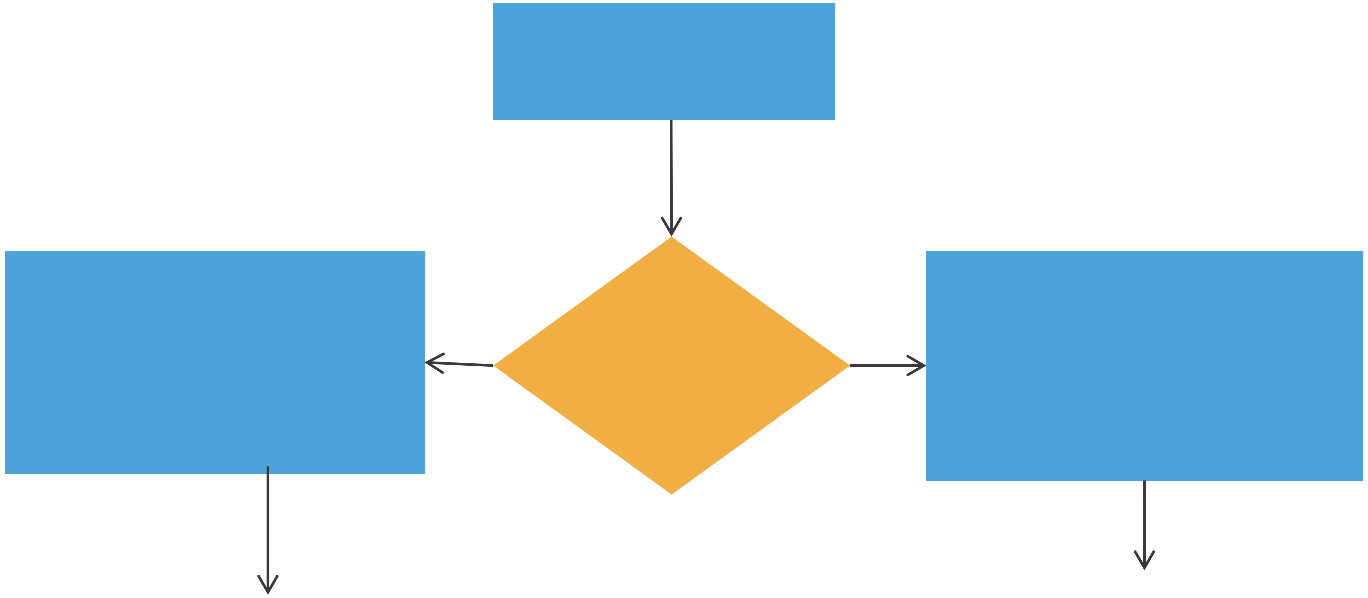
Expanding the Shopping Cart

Criteria for free shipping:

- No change for first 4 lines, no change to the *print* line
- Final price changes if total is $100 or less
- If the product costs $100 or less, we must add shipping cost to the final price the customer pays
- If the product is more than $100, then final price will NOT include a shipping cost

```
productCost = prompt("Enter price of product: ");
productCost = parseFloat(productCost);
TAX = 1.08;
SHIPPING = 5.00;

finalPrice = (productCost * TAX) + SHIPPING;

printOut  = finalPrice;
```

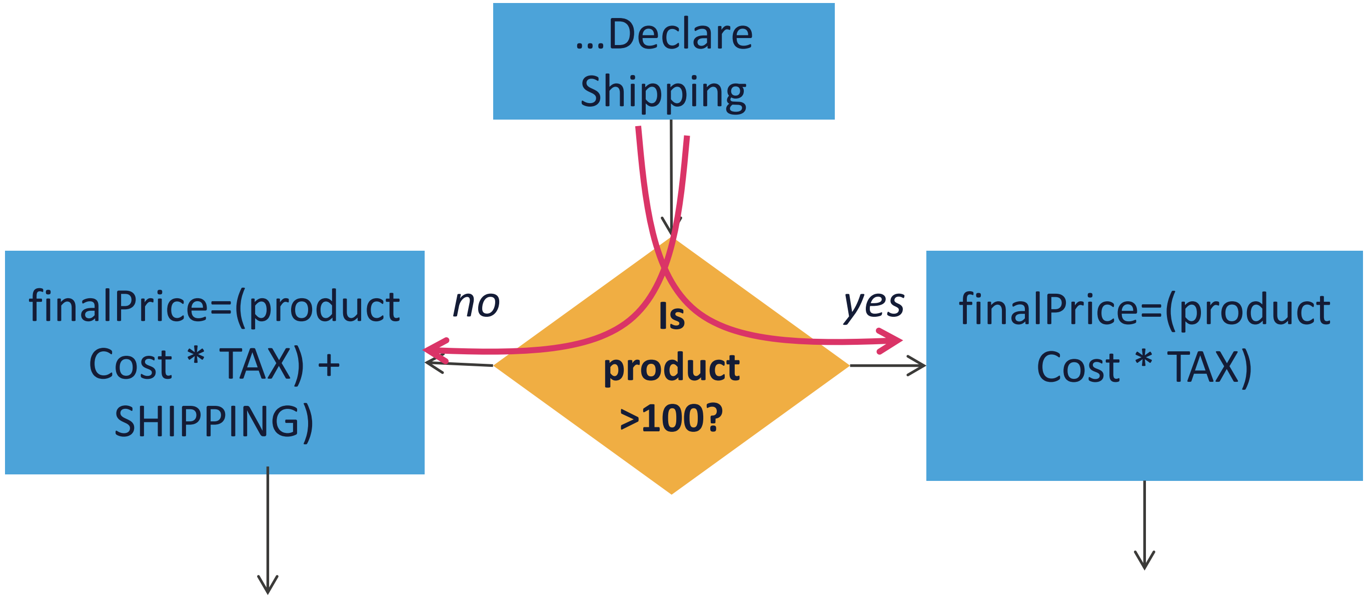
Flowchart for decisions



Diamonds are for Decisions

The diamond represents a decision or selection process. If the product price is < 100 we exit out from the left otherwise we exit to the right.

Flowchart for decisions

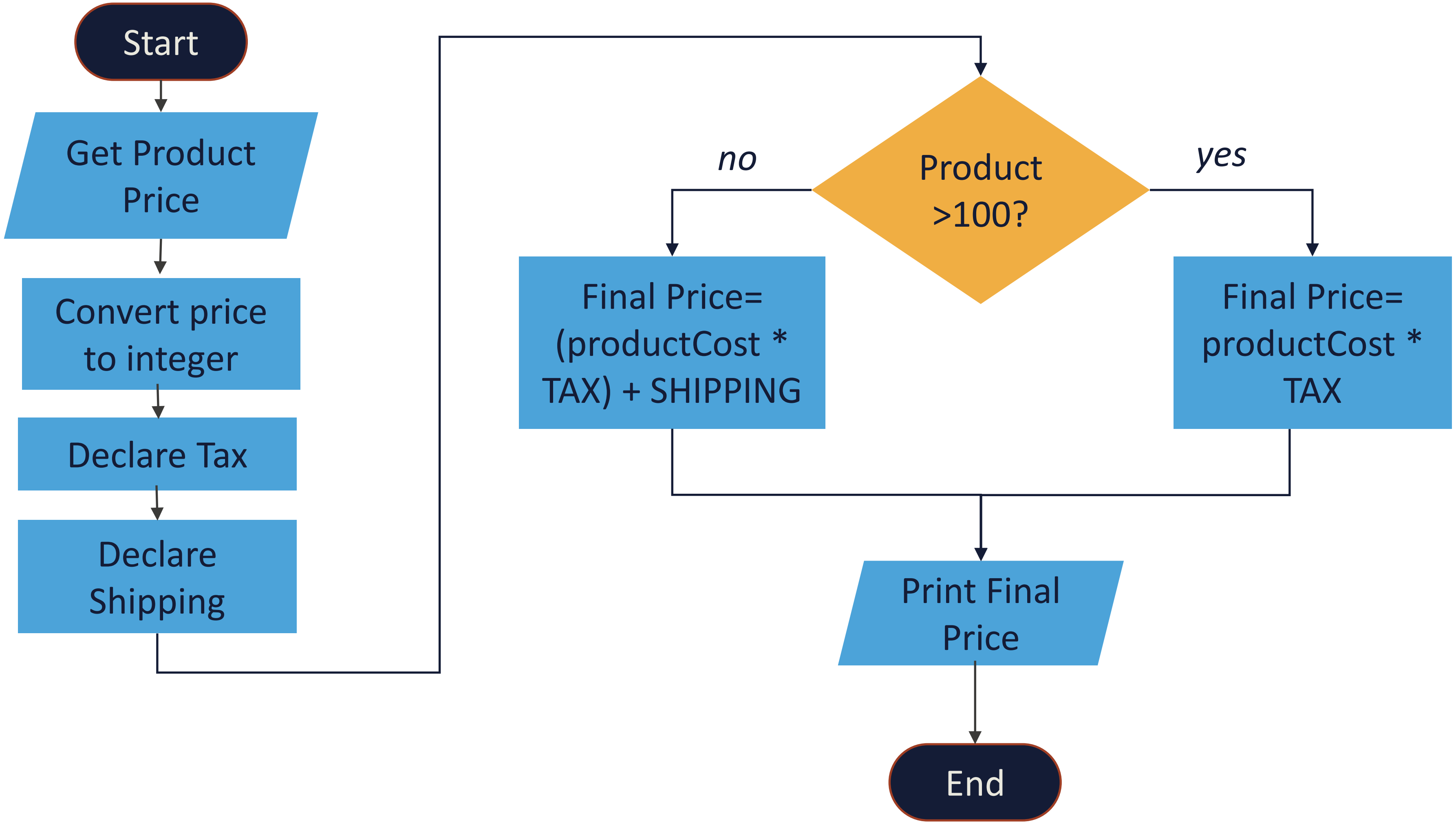

...Declare Shipping
Is product >100?
no
yes
finalPrice=(product Cost * TAX) + SHIPPING)
finalPrice=(product Cost * TAX)

Diamonds are for Decisions

If we answer ‘no’ to the question, then we calculate the final price by multiplying cost by tax amount and we move on to the next step.

However if we answer ‘yes’ then we must ADD shipping cost to the product price times tax. Then we move on to the next step.

Full Program Flowchart



Expanding the Shopping Cart

Selection:

- If they buy more than $100, we calculate the final price just like we have been doing so far
- If they buy less than 100, then we charge a shipping cost, so we add that amount to the final price

```
productCost = prompt("Enter price of product: ");
productCost = parseFloat(productCost);
TAX = 1.08;
SHIPPING = 5.00;
if (productCost > 100){
   finalPrice = productCost * TAX}
else{
   finalPrice = (productCost * TAX) + SHIPPING}
printOut  = finalPrice;
```

Options

Consider these options

- If you have less than 2 years programming experience, you are considered a **junior** programmer
- However, if you have less than 6 years programming (but more than 2), then you are considered **intermediate**
- It means then that more than 6 years qualifies you to be a **senior** programmer

Algorithm

Steps performed by machine:

```
Prompt user for years of experience
Store that value in years
Convert years to a numeric value
If years < 2
  Print Junior
Else If years < 4
  Print Intermediate
Else
  Print Senior
```

- JavaScript, like most languages solve this problem with the *else if* keyword
- Keywords are words that are reserved only for use by the computer language and programmers are not allowed to use them to name variables
- Programers can use them but they must be joined to some other phrase like ifSold or myArray
- Notice the indentation, this is convention NOT a JavaScript requirement

JavaScript Program

Steps performed by machine:

```
let level="";
let years = input("Enter number of years programming JavaScript: ")
let years = int(years);
if (years < 2)
  level = "Junior Programmer“;
else if (years < 4)
  level = "Intermediate Programmer“;
else
  level = "Senior Programmer“;
```

Here is the actual JavaScript program

- All the lines from the algorithm were directly translated into JavaScript code
- However it does not always happen this way

Compound Situations

- 21 and older
- Gathering bubble must be 20 or less
- If you make between 20K and 40K your tax rate is 16%
- You must be shorter than 185cm to ride the scrambler

Compound Comparisons

Steps to be performed by machine:

Prompt user for years of experience and store that value in *years*

```
If years < 2
  Print Junior
Else If years > 2 but < 4
  Print Junior Intermediate
Else if years > 4 but less than 7
  Print Intermediate
Otherwise
  print Senior Programmer
```

- Same program but using compound comparison operators

- Compound comparisons are just like algebra
  - Greater than or equal to
  - Less than or equal to
  - NOT equal to
  - Equality

Consider this problem

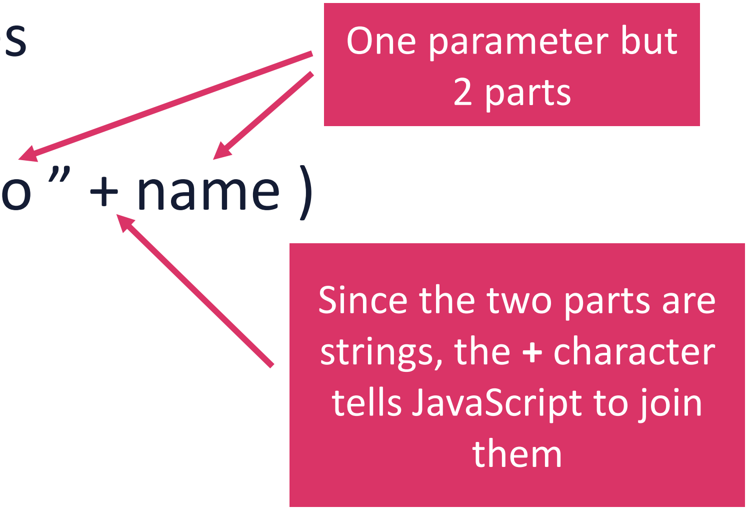
- If you are age 56 or older, you are considered a *Baby Boomer*
- If you are younger than 56 but older than 40 you are *Gen X*
- Younger than 40 but older than 24 you may be *Gen Y*
- Younger than 24 you are *Generation Z*

JS Functions

- A function (method) performs one or more tasks
- alert (“hello”)
- “hello” is a parameter

JS Functions

- Several sub-functions can take place inside the parentheses
- alert (“hello ” + name)



One parameter but 2 parts

Since the two parts are strings, the **+** character tells JavaScript to join them

JS Functions

- JavaScript processes everything inside the parentheses

- let name = "Axle";
- alert("Hello " + name + ", have a nice day :) ");

Some functions coming up

- parseInt()
- parseFloat()
- length()
- toString()
- concat()
- indexOf()
- slice()
- split()
- isNaN()
- **We can write our own functions (soon)**